CLAIMS

What is claimed is:

1. A system for distributed convolution of stacked digital video data comprising:
   a plurality of video data convolve units connected in a chain, wherein a video data convolve unit is operable to:
   receive video pixel data from a video output of a dedicated rendering unit;
   calculate partial convolution sums for a set of the video pixels that are located within a convolution kernel;
   receive accumulated partial convolution sums from a prior video data convolve unit in the chain, unless the video data convolve unit is the first video data convolve unit in the chain;
   add the calculated partial convolution sums to the previously accumulated partial convolution sums; and
   output new accumulated partial convolution sums to the next video data convolve unit in the chain, unless the video data convolve unit is the last video data convolve unit in the chain.

2. The system of claim 1, further comprising one or more partial results buses, wherein each bus connects a video data convolve unit in the chain to a next video data convolve unit in the chain.

3. The system of claim 1, further comprising an interface between the dedicated graphics rendering unit and the video data convolve unit to convert the format of the video pixel data to a digital data format utilized by the video data convolve unit.

4. The system of claim 1, wherein the partial convolution sums comprise (for each parameter value specified for each pixel) 1) a sum of weights determined for locations of each video pixel in the set of video pixels and 2) a sum of weighted video pixel values for the set of video pixels.

5. The system of claim 1, wherein the video data convolve unit comprises a video line buffer utilized to store lines of video pixels received from the video output of the rendering unit.

6. The system of claim 5, wherein the video data convolve unit further comprises a convolution calculation unit that is operable to calculate partial convolution sums for the set of pixels, a partial results accumulator that is operable to add the partial convolution sums to corresponding partial results received and to output the new accumulated partial results, and a pixel value calculator that is operable in the last video data convolve unit in the chain to determine values for a convolved pixel from the final accumulated partial sums.

7. A system for convolution of tiled digital video data comprising:
a plurality of video data convolve units connected in a chain, wherein each unit is operable to:
receive video pixels from a video output of a graphics rendering unit; and
convolve a set of the video pixels that are located within a convolution kernel to determine values for a convolved video pixel corresponding to the convolution kernel, wherein the value for each pixel parameter equals a sum of weighted video pixel values for the parameter divided by a sum of weights, wherein the weights are determined for locations of each video pixel in the set of video pixels.

8. The system of claim 7, further comprising one or more partial video buses, wherein each bus connects a video data convolve unit in the chain to a next video data convolve unit in the chain.

9. The system of claim 7, wherein the video data convolve unit comprises a video blend unit that is operable to receive convolved video pixels from a prior video data convolve unit and output a stream of convolved video pixels that is a combination of the received and generated video pixels ordered by screen location.

10. A system for distributed convolution of stacked and tiled digital video data comprising:

a plurality of video data convolve units connected in a chain, wherein the video data convolve units are separated into a plurality of groups, wherein at least one of the groups has a plurality of video data convolve units, and wherein each group is operable to determine values for convolved video pixels located in a portion of screen space assigned to the group; and

wherein each video data convolve unit within a group is operable to:

receive video pixels from a video output of a dedicated rendering unit;

calculate partial convolution sums for a set of the video pixels that are located within a convolution kernel corresponding to a convolved pixel location;

receive accumulated partial convolution sums from a prior video data convolve unit in the chain, unless the video data convolve unit is the first video data convolve unit within the group;

add the calculated partial convolution sums to the accumulated partial convolution sums; and

output new accumulated partial convolution sums to the next video data convolve unit in the chain, unless the video data convolve unit is the last video data convolve unit within the group.

11. The system of claim 10, further comprising one or more partial video buses, wherein each bus connects a last video data convolve unit in a group to a last video data convolve unit in the next group in the chain of video data convolve units.

12. The system of claim 10, further comprising one or more partial results buses, wherein each bus connects a video data convolve unit to a next video data convolve unit in a group.

13. The system of claim 10, wherein a video data convolve unit comprises a video blend unit, and wherein a last video data convolve unit in a group is operable to receive convolved video pixels from a prior group's last video data convolve unit

and output a stream of convolved video pixels that is a combination of the received and generated convolved video pixels ordered by screen location.

14. A method for distributed convolution of stacked digital video data in a plurality of video data convolve units connected in a chain comprising (for each video data convolve unit):
receiving video pixel data from a video output of a dedicated rendering unit;
storing the video pixel data in a video line buffer;
performing a partial convolution as part of a distributed process to determine values for a convolved pixel by calculating partial convolution sums for the pixels in the line buffer that are located within a convolution kernel corresponding to the location of a convolved pixel;
adding the partial convolution sums to any corresponding accumulated partial convolution sums received from a prior video data convolve unit in the chain to form new accumulated partial convolution sums, unless the video data convolve unit is the first video data convolve unit in the chain; and
sending the new accumulated partial convolution sums to the next video data convolve unit in the chain, unless the video data convolve unit is the last video data convolve unit in the chain.

15. The method of claim 14, further comprising:
specifying a different jitter value or jitter pattern for each rendering unit;
sending vertex data for each geometric primitive to each rendering unit;
rendering pixel values for each jittered pixel location that lies within a geometric primitive; and
outputting the pixel values.

16. The method of claim 14, further comprising for the last video data convolve unit in the chain: determining parameter values for a convolved pixel from the final accumulated partial convolution sums, storing the convolved pixel values in a video output buffer, and outputting the convolved pixel data.

17. The method of claim 16, wherein determining parameter values for a convolved pixel comprises (for each parameter) dividing the final accumulated sum of weighted video pixel values for the parameter by a sum of weights, wherein the weights are determined for locations of each video pixel that is within the convolution kernel.

18. The method of claim 14, further comprising interfacing the video data output from the rendering unit to the digital data format utilized in the video data convolve unit.

19. The method of claim 14, wherein the partial convolution sums comprise 1) a sum of weights determined for locations of each video pixel in the set of video pixels and 2) a sum of weighted pixel values for the set of video pixels.

20. The method of claim 14, wherein the video pixel data from each rendering unit are determined for primitives that are geometrically expanded in both x and y dimensions by an integer factor of 2 or more; and wherein convolved pixel values are determined from the geometrically expanded pixel data and then assigned to convolved pixel locations determined by reducing the expanded locations by the same integer factor.

21. The method of claim 14, wherein each graphics rendering unit renders video pixels for primitives located anywhere in screen space.

22. A method for convolution of tiled digital video data in a plurality of video data convolve units connected in a chain comprising (for each video data convolve unit):
receiving video pixel data from a video output of a dedicated rendering unit for a specified portion of screen space;
storing the video pixel data in a video line buffer;

determining values for a convolved pixel by convolving the video pixels in the line buffer that are located within a convolution kernel corresponding to the location of a convolved pixel;

storing the convolved pixel in a convolved pixel buffer;

combining the convolved pixels in the pixel buffer with convolved pixels received from a prior video data convolve unit so that the combined convolved pixels are ordered by their locations in a line of screen space, unless the video data convolve unit is the first video data convolve unit in the chain; and

sending the combined convolved pixels to the next video data convolve unit in the chain, unless the video data convolve unit is the last video data convolve unit in the chain.

23. The method of claim 22, wherein a last video data convolve unit in the chain outputs the combined and ordered convolved video pixels to a display.

24. The method of claim 22, wherein each rendering unit renders video pixels for a different portion of screen space.

25. The method of claim 22, wherein frustum culling may be utilized to sort the geometric primitives by screen portions and send each primitive to the corresponding rendering unit, and wherein those primitives that overlap a boundary between two screen portions may be sent to both corresponding rendering units or subdivided along the boundary.